Multi-class Protein Fold Recognition Using Support Vector Machines and Neural Networks

Chris H.Q. Ding* and Inna Dubchak
NERSC Division, Lawrence Berkeley National Laboratory
University of California, Berkeley, CA 94720, USA

Abstract

Motivation: Protein fold recognition is an important approach to structure discovery without relying on sequence similarity. We study this approach with new multi-class classification methods and examined many issues important for a practical recognition system.

Results: Most current discriminative methods for protein fold prediction use the one-against-others method, which has the well-known "False Positives" problem. We investigated two new methods: the unique one-against-others and the all-against-all methods. Both improve prediction accuracy by 14-110% on a dataset containing 27 SCOP folds. We used the Support Vector Machine and the Neural Network learning methods as base classifiers. SVM converges fast and leads to high accuracy. When scores of multiple parameter datasets are combined, majority voting reduces noise and increases recognition accuracy. We examined many issues involved with large number of classes, including dependencies of prediction accuracy on the number of folds and on the number of representatives in a fold. Overall, recognition systems achieve 56% fold prediction accuracy on a protein test dataset, where most of the proteins have below 25% sequence identity with the proteins used in training.

Contact: chqding@lbl.gov, ildubchak@lbl.gov

Supplementary Information: The protein parameter datasets used in this paper is available online (http://www.nersc.gov/~cding/protein).

Keywords: protein fold recognition, protein structure, multi-class classification, support vection machines, neural networks.

*To whom correspondence should be addressed.

1 Introduction

Computational analysis of biological data obtained in genome sequencing and other projects is essential for understanding cellular function and the discovery of new drugs and therapies. Sequence-sequence and sequence-structure comparison play a critical role in predicting a possible function for new sequences. Pairwise sequence alignment is accurate in detecting close evolutionary relationship between proteins (Holm & Sander 1999), but it is not efficient when two proteins are structurally similar, but have no significant sequence similarity. The threading approach has demonstrated promising results in detecting the latter type of relationship (Jones 1999).

In this paper, we focus on the taxonometric approach in determining structure similarity without sequence similarity, using machine learning methods (Baldi & Brunak 1999, Durbin, et al, 1998). such as Neural Networks and Support Vector Machines. This approach has achieved some success mostly through recognition of the protein fold, which is a common 3-dimensional pattern with the same major secondary structure elements in the same arrangement and with the same topological connections (Craven at al 1995). The taxonometric approach presumes that the number of folds is restricted and thus the focus is on structural predictions in the context of particular classification of 3D folds. Detailed, comprehensive protein classifications such as SCOP (LoConte et al. 2000) and CATH (Pearl et al 2000) identified more than 600 3D protein folding patterns. Protein fold prediction in the context of this large number of classes presents a rather challenging classification problem. The more classes are involved, the more difficult it is to accurately predict the fold for a query sequence.

Most current studies use the one-vs-others (one-against-others) method, which clearly does not scale well to a large number of classes due to the complexity of the "others" classes (Chou & Zhang 1995, Dubchak et al. 1995). For these reason, we studied two improved methods: the unique one-vs-others, and the all-vs-all methods. However, these new methods, essentially based on all pairs of individual classes, require building very large number of discriminative classifiers, (about 84,000 in our database of 27 folds). We overcome this difficulty by using the newly developed Support Vector Machine.

Support Vector Machine (SVM) is a new discriminative method (Vapnik 1995), which has demonstrated high classification accuracy in protein family (evolutionary relationship) prediction (Jaakkola et al 1999), gene expression classification (Brown et al, 2000), and many other areas beyond molecular biology. An advantage of SVM is its fast convergence in training, about 10-100 faster than in Neural Network (as described later). Thanks to fast speed in SVM training, we were able to carry out systematic investigation on the three multi-class classification methods. We also carried out fold prediction using NN with the new multi-class recognition methods. Comparison of NN with SVM provided new insights into these learning methods.

2 Multi-class Prediction Methods

Many discriminative methods, including SVM and NN, are often most accurate and efficient when dealing with two classes only (they can deal with more classes, but usually at reduced accuracy and efficiency). For large number of classes, higher-level multi-class methods are developed that utilize these two-class classification methods as the basic building blocks.

2.1 One-vs-Others Method

This is a simple and effective method (Dubchak et al. 1999, Brown et al, 2000) for multi-class problems. Suppose there are K classes in the problem. We partition the K classes into a two-class problem: one class contains proteins in one "true" class, and the "others" class combines all other classes. A two-class classifier is trained for this two-class problem. We then partition the K classes into another two-class problem: one class contains another original class, and the "others" class contains the rest. Another two-way classifier is trained. This procedure is repeated for each of the K classes, leading to K two-way trained classifiers.

In the recognition process, the system tests the new query protein against each of the K two-way classifiers, to determine if it belongs to the given class or not. This leads to K scores from the K classifiers. Ideally, only one of the K classifiers will show a positive result and all other classifiers show negative results, assigning the query protein to a unique fold. In practice, however, many proteins show positive on more than one class, leading to ambiguous prediction results, the so-called "False Positive" problem. One of the main reasons for the false positive problem is that the decision boundary between one "true" class and its complementary combined "others" class cannot be drawn cleanly, due to the complexity of the "others" class and close parameter proximity of some proteins.

2.2 Unique One-vs-Others Method

Here we propose a new method to improve upon the one-vs-others method. The idea is to obtain an unambiguous prediction for a given query protein sequence. This is achieved by reducing or eliminating false positives. We add a second step to the one-vs-others method by applying two-way discriminative classifications on the pairs between all the classes with positive predictions. Suppose for a query protein the one-vs-others system predicts 4 positives, i.e., 4 folds. There are 6 possible pairs out of these 4 folds. A 2-way classifier is trained for each of the pairs, is applied to the query protein, and produces a positive prediction (vote) for a particular fold. All votes from the 6 classifiers are tallied and the class with the most votes represents the final prediction. Therefore the false positive problem is eliminated at this second step (see example in section 6).

Note that in the false positives elimination step, the decision boundary is drawn between two "true" classes of training proteins, instead of between one "true" class and its complementary "others" class, which is highly complex. Thus false positives are eliminated accurately. Therefore, the unique one-vs-others method has higher prediction accuracy.

The false positives elimination step is essentially a noise reduction technique. We expect it to work particularly well for classification methods such as Neural Networks which have large false positive rates or noise (see Section 5). Indeed, we found in our experiments that it reduces the error rates of Neural Networks by almost a factor of 2.

2.3 All-vs-All Method

In the unique one-vs-others method, after obtaining the results of the one-vs-others method, two-way classifiers between two "true" classes are trained and used to break "ties" between multiple positives including both true and false positives. We can generalize this further and eliminate the one-vs-others method entirely. This method therefore depends entirely on two-way classifiers between pairs of "true" classes, and achieves higher accuracy in the resulting classifications.

In this method, we train two-way classifiers between all possible pairs of classes; there are K(K-1)/2 of them. A new query protein is then tested against these K(K-1)/2 classifiers and obtains K(K-1)/2 positive scores (votes). In a perfect case, the correct class will get the maximum possible votes, which is K-1 for all class-class pairs; and votes for other K-1 classes would be randomly distributed, leading to [K(K-1)/2 - K-1]/(K-1) = (K-2)/2 per class on average. Thus we expect an average *signal-to-noise ratio* of

$$r = 2(K-1)/(K-2) \simeq 2,$$

a fairly large margin. Furthermore, the output class is unique: for any query sequence, there can only be one class that gets the maximum possible vote.

In practice, however, the number of votes for each protein has large variations. The most popularly voted class do not necessarily get the maximum possible number of votes; the number of votes for each class tends to decrease gradually from maximum to minimum, i.e., the margin between the correct class and incorrect classes is not as large as K-1 vs (K-1)/2 in the above analysis. For this reason, our voting method simply outputs the class with the highest vote, regardless of whether this vote is a maximum possible vote or not.

A problem with both all-vs-all and unique one-vs-others methods is the large number of 2-class classifiers required. However, for the 600 folds in SCOP database, this task can be easily handled using current computers. The SVMs can be trained in reasonable time (1-2 days on a workstation), and the estimated memory requirement is about 1GB.

3 Two-class Classifications

The three multi-class classification methods above utilize 2-class classifiers as their building blocks, which are described below.

3.1 Support Vector Machine

SVM is a new and promising binary classification method developed by Vapnik and colleagues at Bell Laboratories (Vapnik 1995, Burges 1998), with algorithm improvements by others (Osuna, 1997, Joachims 1998). SVM is a margin classifier. It draws an optimal hyperplane in a high-dimensional feature space (determined by $\mathbf{w}, b$); this defines a boundary that maximizes the margin between data samples in two classes, therefore giving good generalization properties. The decision boundary is defined by the function

$$f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b$$

Depending upon the sign of the function, protein $\mathbf{x}$ is classified into either of the two classes. For many problems where samples of different classes cannot be separated in the original feature space, one can effectively embed the problem in higher dimensional space (indicated by $\phi(\mathbf{x})$), making it easier to find the optimal hyperplane, i.e., better decision boundary. The actual embedding is achieved through a kernel function, making it easy to implement and fast to compute.

For our datasets, we found that linear kernel does not work well, the polynomial kernel works better, and the Gaussian kernel gives the best results. To account for the imbalance of positive examples (proteins in a "true" class) and the negative examples (those in the "others" class), in the one-vs-others method, we duplicate the positive examples to approximately match the number of negative examples. This works out well. In all the 2-class training sessions (about 84,000), not a single protein is misclassified.

3.2 Neural Network

We used three-layer feed-forward neural networks with the weights adjusted by conjugate gradient minimization. Since in NN training there is always a problem of generalization, the number of NN parameters was adaptively adjusted to variable training set sizes by changing the number of hidden units. Various NN architectures were tested; the geometry ($N_{\text{hidden}} = 1$ and $N_{\text{out}} = 2$) achieves a good performance while having a minimum overall number of nodes (to improve generalization). We found it adequate for the recognition of all folds in the database. The number of inputs is the same as the dimensionality of the feature vectors. High activity output to one node indicated the assignment of the test sequence to a particular fold, and high activity to the other node indicated the assignment to the other folds.

Fold	Index	N_{train}	N_{test}
α :			
Globin-like	1	13	6
Cytochrome c	3	7	9
DNA-binding 3-helical bundle	4	12	20
4-helical up-and-down bundle	7	7	8
4-helical cytokines	9	9	9
Alpha; EF-hand	11	7	9
β :			
Immunoglobulin-like β-sandwich	20	30	44
Cupredoxins	23	9	12
Viral coat and capsid proteins	26	16	13
ConA-like lectins/glucanases	30	7	6
SH3-like barrel	31	8	8
OB-fold	32	13	19
Trefoil	33	8	4
Trypsin-like serine proteases	35	9	4
Lipocalins	39	9	7
α/β :			
(TIM)-barrel	46	29	48
FAD (also NAD)-binding motif	47	11	12
Flavodoxin-like	48	11	13
NAD(P)-binding Rossmann-fold	51	13	27
P-loop containing nucleotide	54	10	12
Thioredoxin-like	57	9	8
Ribonuclease H-like motif	59	10	14
Hydrolases	62	11	7
Periplasmic binding protein-like	69	11	4
$\alpha + \beta$:			
β-grasp	72	7	8
Ferredoxin-like	87	13	27
Small Inhibitors, toxins, lectins	110	12	27

Table 1: Non-redundant subset of 27 SCOP folds used in current study

4 Dataset

4.1 Training Dataset

The dataset we used for training was selected from the database built for the prediction of 128 folds in our earlier study (Dubchak, et al., 1999). This database was based on the PDB_select sets (Hobohm, et al, 1992, Hobohm and Sander, 1994) where two proteins have no more than 35% of the sequence identity for the aligned subsequences longer than 80 residues. Since the accuracy of any machine learning method depends directly on the number of representatives for training, we utilized 27 most populated folds in the database which have seven or more proteins and represent all major structural classes: α, β, α/β, and $\alpha+\beta$. The folds in our database and the corresponding number of proteins in training (N_{train}) are shown in Table 1.

Symbol	Parameter	Dim
C	amino acids composition	20
S	predicted secondary structure	21
H	hydrophobicity	21
V	normalized van der Waals volume	21
P	polarity	21
Z	polarizability	21

Table 2: Six parameter datasets extracted from protein sequence. The dimension of the feature vector are also shown.

4.2 Independent Test Dataset

As an independent dataset for testing we used the PDB-40D set developed by the authors of the SCOP database (Lo Conte 2000). This set contains the SCOP sequences having less than 40% identity with each other. From this set we selected 386 representatives of the same 27 largest folds (N_{test}) shown in Table 1. All PDB-40D proteins that had higher than 35% identity with the proteins of the training set were excluded from the testing set.

4.3 Feature Vector Extraction

To use machine learning methods, feature vectors are extracted from protein sequences. Percentage composition of the 20 amino acids forms a parameter set. For each structural or physico-chemical property listed in Table 2, feature vectors are extracted from the primary sequence based on three descriptors: "composition," percent composition of 3 constituents (e.g, polar, neutral and hydrophobic residues in hydrophobicity); "transition," the transition frequencies (polar to neutral, neutral to hydrophobic, etc.); and "distribution," the distribution pattern of constituents (where the first residue of a given constituent is located, and where 25%, 50%, 75% and 100% of that constituent are contained). For concrete details, see (Dubchak et al., 1995, 1999). The entire feature datasets are available on line (http://www.nersc.gov/~cding/protein). With the feature extraction method, feature vectors (we call them parameter vectors) can be easily calculated from new protein sequences, and fold prediction by different machine-learning techniques can be performed rapidly and automatically.

Note that the six feature vector datasets (parameter sets) are extracted independently. Thus, one may apply machine-learning techniques based on a single parameter set for protein fold prediction. We found that using multiple parameter sets and applying majority voting on the results lead to much better prediction accuracy. This is the approach we take in this study. Alternatively, one may combine different parameter sets into one dataset so that each protein is represented by a 125-dimensional feature vector. We experimented with this approach and found that the prediction accuracy is not enhanced.

5 Accuracy Measure

Assessing the accuracy of various discriminative methods so far mostly involves calculating true positive rates (TPR) and false positives rates (FPR). These characteristics are originally designed for two-class problems, closely related to sensitivity and selectivity used in sequence comparison methods (Brenner et al. 1998); they are now extended to problems involving more than two classes, through the one-vs-others method (e.g., Dubchak et al,1999, Jaakkola et al, 1999). However, multi-way classification methods are not restricted to the one-vs-others method. The all-vs-all method discussed above is another example. In these methods, there are no such concepts as true positives and false positives. Therefore we need an accuracy measure which can deal with all situations.

In this paper, we use the standard Q percentage accuracy (Rost & Sander, 1993, Baldi et al, 2000), generalized to handle true positives and false positives. Suppose we have $N = n_1 + n_2 + \cdots + n_K$ test proteins (n_1 are observed to belong to class F1, etc.). Suppose that out of n_1 proteins, c_1 are correctly and *uniquely* recognized as belonging to F1, etc., so that total $C = c_1 + c_2 + \cdots + c_K$ proteins are correctly recognized (c_i's correspond to diagonal entries in the $K \times K$ contingency table). The accuracy for class i is $Q_i = c_i/n_i$. The overall or total accuracy is $Q = C/N$ ($Q = Q_{\text{total}}$).

Individual Q_i relates to the overall Q in a very simple way. An individual class contributes to the overall accuracy in proportion to the number proteins in its class, and thus has a weight $w_i = n_i/N$. Therefore the overall accuracy equals the weighted average over individual classes:

$$Q = \sum_{i=1}^{K} w_i q_i = C/N.$$

Here "unique" means that a single fold is predicted for an unknown protein. False positives are taken into account by considering them as "ties." If a protein is tested positive for 4 classes, and one of them is correct, then $c = 1/4$ for this protein. So in general, c_i are not necessarily integers. Allowing fractions in the contingency table, the one-against-others method can be properly accommodated. (Conventional contingency tables are defined to have integer entries only.) We sometimes call this generalized accuracy definition *unique* accuracy, when it's applied to the one-vs-others method.

We can similarly define $\text{TPR}_i = \text{TP}_i/n_i$, and $\text{FPR}_i = \text{FP}_i/n_i$, for each class i, and overall TPR and FPR as the weighted average. The differences between unique accuracy and TPR and FPR are illustrated in Table 3. Here we used three different 2-way classifiers: the SVM1 (less optimized), SVM2 (better optimized) and NN. In general, for a given method, the higher TPR it achieves, the higher the FPR it brings, as we move from SVM1 to SVM2 to NN. However, the unique accuracy clearly indicates SVM2 is the best among the three methods: SVM2 achieves most correct unique recognitions, even though SVM2 has a higher FPR than SVM1, and SVM2 has a lower TPR than NN.

Method	TPR	FPR	Q
SVM1	33.8 %	7.5 %	33.5 %
SVM2	48.8	48.6	43.5
NN	59.5	296	20.5

Table 3: Overall TPR, FPR and unique accuracy (Q) of three classification methods using one-vs-others classification method, on the independent test set using the composition dataset only. Results are weighted averages over the 27 folds.

For NN, although its TPR of 59.5% is quite high, the large FPR of 296% brings the unique accuracy down to only 20.5%. For SVM1, although its FPR of 7.5% is low, the low TPR of 33.8 % cannot move up the unique accuracy.

6 Tests on Independent Datasets

Once the recognition system is built and trained, we can test it in two ways. In the first test, we test the system against a dataset which is independent of the training dataset. Note that test proteins have less than 35% sequence identity with those used in training.

6.1 One-vs-Others Method

In this method, on the datasets with 27 fold classes, we build 27 two-way one-vs-others classifiers, with either SVM or NN. Each protein in the test set is tested against all 27 two-way classifiers. If the result is positive, this is a positive vote for the class. However, if the result is negative, i.e., the protein belongs to one of the 26 other classes, or equivalently, the protein belongs to each of the other 26 classes with a probability of 1/26. But these small fractional votes will not change the results discussed below, and is negligible for large number of classes. An example of protein `1hbg` (using NN with composition dataset only) is :

```
1hbg   (F1)  1:1 46:1 47:1 51:1
```

Here (F1) indicates that `1hbg` belongs to class F1; 1:1 indicates 1 positive vote for class F1; 46:1 indicates 1 positive for class F46, etc. Protein `1hbg` has 4 positives.

We can combine votes obtained from different parameter sets to improve prediction accuracy. For example, when votes of all 6 parameter datasets are combined, we have,

```
1hbg  (F1) 1:6 20:2 46:2 47:2 51:2 4:1 7:1 9:1 11:1 30:1 32:1
```

Protein `1hbg` now has 6 votes for class F1, 2 votes for classes F20, F46, F47, etc. Thus 6 parameter sets improve the accuracy of `1hbg` from 4 positives to 1, a unique correct positive. Although the majority of proteins benefit from combining multiple votes, there are some exceptions, reflecting the statistical nature of these methods.

Fold	C	CS	CSH	CSHP	CSHPV	ALL6
1	19.6%	68.2%	70.8%	63.9%	66.7%	55.6%
3	9.6	37.0	44.8	48.1	43.3	27.8
4	11.3	27.8	23.2	24.9	25.7	25.6
7	12.3	21.8	39.6	46.3	37.5	37.5
9	25.4	83.3	88.9	75.0	80.6	77.8
11	13.3	28.9	47.5	44.4	38.9	27.8
20	16.0	32.3	37.4	44.1	45.5	53.9
23	23.2	17.5	11.0	12.5	14.2	12.5
26	20.6	25.5	30.5	37.4	43.8	44.2
30	13.9	29.2	35.4	33.3	33.3	33.3
31	14.8	26.8	33.8	40.6	45.8	52.1
32	9.4	20.6	22.8	18.4	25.4	26.3
33	18.8	25.0	37.5	29.2	33.3	25.0
35	12.5	11.9	14.6	18.8	18.8	0.0
39	24.3	27.9	31.4	42.9	35.7	40.5
46	38.7	58.1	72.4	69.1	65.8	65.8
47	21.9	54.2	40.5	34.2	33.3	38.9
48	12.1	25.5	25.6	15.4	17.9	21.8
51	24.8	38.6	40.2	41.4	41.0	42.6
54	14.5	24.8	20.8	22.2	27.8	29.2
57	18.1	31.2	32.5	38.5	37.9	50.0
59	21.4	37.0	41.1	38.1	39.3	38.1
62	47.6	42.5	41.8	47.6	42.9	57.1
69	0.0	0.0	0.0	0.0	0.0	0.0
72	8.0	9.4	12.5	16.7	16.7	25.0
87	9.0	28.3	35.6	33.6	28.7	21.4
110	32.7	58.9	54.1	58.9	61.2	60.3
avg	20.5	36.8	40.6	41.1	41.2	41.8

Table 4: Unique Accuracy Q_i for each fold class and overall Q (bottom line), for the one-vs-others method using neural networks. Votes are combined gradually, with the order "C", "S", "H", "P", "V", "Z" (see Table 2).

Table 4 shows results of this combination of votes using NN for each individual fold. As the number of parameter sets increases, the prediction accuracy for most classes increases steadily, although not uniformly, reflecting the statistical nature of the prediction system. The overall prediction accuracy increases very substantially, from 20.5% for the composition set alone (denoted as C) to 36.8% for composition+secondary datasets (denoted as CS), to 40.6% for composition+secondary + hydrophobicity (denoted as CSH).

The reason for this is noise reduction. NN has rather high true positive rates of 59.5% (Table 3), but also has high false positive rate 296.0%, so each proteins has about 3 false positives. The high FPR brings the unique accuracy down to 20.5%. When scores of different parameter sets are combined, the majority voting helps to reduce the false positives, and thus improves the final unique accuracy.

	C		S		CSH	
Fold	OvO	uOvO	OvO	uOvO	OvO	uOvO
1	75.0	83.3	41.7	50.0	87.5	83.3
3	44.4	55.6	16.7	33.3	50.9	66.7
4	34.2	35.0	36.7	40.0	43.7	46.7
7	43.8	50.0	35.4	29.2	53.5	62.5
9	94.4	100.0	44.4	55.6	69.8	100.0
11	33.3	44.4	27.8	22.2	50.0	55.6
20	41.3	52.3	36.0	36.4	48.6	60.2
23	16.7	33.3	8.3	11.1	15.3	16.7
26	46.2	38.5	10.3	30.8	46.8	53.8
30	33.3	33.3	16.7	16.7	25.0	33.3
31	54.2	62.5	37.5	37.5	41.7	50.0
32	21.1	21.1	22.4	22.8	27.4	31.6
33	50.0	50.0	37.5	50.0	50.0	50.0
35	50.0	50.0	25.0	25.0	25.0	25.0
39	42.9	42.9	28.6	28.6	39.3	50.0
46	58.0	66.7	42.2	46.4	60.5	64.6
47	50.0	50.0	66.7	75.0	56.9	54.2
48	33.3	30.8	23.1	30.8	29.5	34.6
51	46.3	55.6	22.2	24.1	31.2	46.9
54	50.0	41.7	33.3	37.5	47.2	36.1
57	18.8	37.5	25.0	25.0	25.0	25.0
59	35.7	35.7	35.7	50.0	39.3	28.6
62	71.4	71.4	50.0	57.1	78.6	71.4
69	25.0	25.0	25.0	25.0	25.0	25.0
72	25.0	25.0	0.0	0.0	25.0	25.0
87	14.8	14.8	16.7	22.2	24.5	29.6
110	67.9	88.9	46.3	55.6	69.3	83.3
avg	43.5	49.4	31.5	36.2	45.2	51.1

Table 5: Unique Percentage Accuracy Q_i, Q for One-vs-Others (OvO) and Unique One-vs-Others (uOvO) methods using Support Vector Machine.

The accuracy for SVM is generally higher than that for NN, because SVM has far less false positives. Table 5 contains the prediction results. The accuracy for SVM is 43.5% on the composition parameter set alone, in contrast to 20.5% for NN. It increases to 45.2% for CSH, in contrast to 41.1% for NN. When more votes for different parameter sets are combined (results shown in Table 6), accuracy improves from 43.5% to 45.2%. This is not as significant as for NN, because false positive rates for SVM are already quite low (see Table 3).

6.2 Unique One-vs-Others Method

Here we eliminate false positives by using two-way discriminative classifications on the pairs between all the classes with positive (both true and false) predictions in the one-vs-others step. For example, protein `1hbg` is voted positive for 4 classes as the result of one-vs-others prediction. We further applied 6 2-way classifiers between the 4 positive classes to `1hbg`, and obtained the following result

```
1hbg  (F1) 1:3 46:1 47:1 51:1
```

The most popularly voted class is now uniquely determined to be F1, and 3 false positives are eliminated.

Results of the uOvO method are shown in Tables 5 and 6. For SVM, the unique one-vs-others (uOvO) method shows good improvements, about 13.6 % for composition data, and 14.9% for secondary structure data. The best final results of the uOvO method are achieved on the combined C+S+H dataset, 51.1%, improved upon the original OvO results of 45.2%. On the NN results (not shown), the average accuracy is improved from 20.5% to 43.1%, a 110% improvement, due to the elimination of the large amount of false positives. These significant improvements indicate the usefulness of the uOvO method in reducing FPR or noise.

6.3 All-vs-all method

For the 27 fold classes, the prediction system consists of $27 * (27 - 1)/2 = 351$ two-way SVM classifiers, each between one pair of folds. A test protein is tested against all trained SVMs, and results are tallied as before. For example, for the protein `1hbg` we get

```
1hbg  (F1) 1:26 46:24 47:24 51:23 3:22 69:21 48:20 35:18 59:18 23:16
```

Folds are sorted according to their votes (more folds with less votes are not shown here). This was repeated for all 6 parameter sets, resulting in a total of 2106 two-way classifiers. The fast convergence of SVM makes this study possible. Due to slow convergence in NN training, training such a large number of NNs would be prohibitive, thus no NN test is done using the all-vs-all method.

Prediction results for the test dataset for each of the folding classes using SVM are shown in Tables 6 and 7. For the composition dataset alone, the unique accuracy is 44.9%. As scores of more parameter datasets are combined together, the accuracy increases to 52.1% for CS, and to 56.0% for CSH due to

	C	CS	CSH	CSHP	CSHPV	ALL6
OvO NN	20.5%	36.8%	40.6%	41.1%	41.2%	41.8%
OvO SVM	43.5	43.2	45.2	43.2	44.8	44.9
uOvO SVM	49.4	48.6	51.1	49.4	50.9	49.6
AvA SVM	44.9	52.1	56.0	56.5	55.5	53.9

Table 6: Unique Accuracy Q for the independent test as more votes on different parameter datasets are combined, for one-vs-others (OvO), unique one-vs-others (uOvO), and all-vs-all (AvA) methods.

	Independent Test				Cross Validation	
Fold Index	NN OvO	SVM OvO	SVM uOvO	SVM AvA	NN OvO	SVM AvA
1	55.6	87.5	83.3	83.3	36.5	73.1
3	27.8	50.9	66.7	77.8	7.1	71.4
4	25.6	43.7	46.7	35.0	33.3	66.7
7	37.5	53.5	62.5	50.0	14.3	42.9
9	77.8	69.8	100.0	100.0	38.9	50.0
11	27.8	50.0	55.6	66.7	21.4	28.6
20	53.9	48.6	60.2	71.6	51.2	46.7
23	12.5	15.3	16.7	16.7	22.2	33.3
26	44.2	46.8	53.8	50.0	28.1	62.5
30	33.3	25.0	33.3	33.3	7.1	21.4
31	52.1	41.7	50.0	50.0	0.0	62.5
32	26.3	27.4	31.6	26.3	7.7	15.4
33	25.0	50.0	50.0	50.0	0.0	12.5
35	0.0	25.0	25.0	25.0	13.3	22.2
39	40.5	39.3	50.0	57.1	11.1	22.2
46	65.8	60.5	64.6	77.1	64.9	82.8
47	38.9	56.9	54.2	58.3	18.2	36.4
48	21.8	29.5	34.6	48.7	13.6	9.1
51	42.6	31.2	46.9	61.1	29.5	53.8
54	29.2	47.2	36.1	36.1	8.3	60.0
57	50.0	25.0	25.0	50.0	25.9	33.3
59	38.1	39.3	28.6	35.7	13.3	5.0
62	57.1	78.6	71.4	71.4	6.8	36.4
69	0.0	25.0	25.0	25.0	34.8	63.6
72	25.0	25.0	25.0	12.5	0.0	0.0
87	21.4	24.5	29.6	37.0	9.0	19.2
110	60.3	69.3	83.3	83.3	55.8	75.0
avg	41.8	45.2	51.1	56.0	27.2	45.4

Table 7: Prediction accuracy Q_i (in percentage) for each individual fold and overall accuracy Q (bottom line). Majority voting is used on combination of votes from different parameter datasets.

noise reduction. In general, the all-vs-all method improves the prediction accuracy by about 24% over the one-vs-others method, and by about 10% over the unique one-vs-others method.

7 Cross-Validation

Another standard test on the recognition system we used was a cross-validation (CV) test. CV measures the performance of the prediction system in a self-consistent way by systematically leaving out a few proteins (about 10%) during the training process and testing the trained prediction system against those left-out proteins. This is repeated such that every protein in the dataset is once among those left-out. Compared to the test on independent set, cross-validation has less bias and better predictive and generalization power.

One such 10-fold cross-validation is run on a random partitioning of a parameter dataset. To gain high statistics, we did four independent partitionings and corresponding CVs. (The total number of 2-way

SVM classifiers trained in this study is 4*S*K*P = 6480 in the one-vs-others method, and 4*S*[K(K-1)/2]*P = 84240 in the all-vs-all method.)

The results of the 10-fold cross-validation are listed in Table 7 for SVM/AvA and NN/OvO. For the composition dataset alone, the CV average unique accuracy is 33%. As scores of more parameter datasets are combined together, the accuracy improves, to 45.4% for C+S+H. By using NN for all 6 parameter sets combined, we achieved an accuracy rate of 27.2%.

8 Summary and Discussions

8.1 Comparison of multi-way classification methods

Our extensive results clearly demonstrate that the two advanced methods, the unique one-vs-others method and the all-vs-all method, outperform the popular one-vs-others method: they improve prediction accuracy by about 14-25 % for SVM, and by about 110% for NN. Of course, the substantial advantages of the advanced methods come at the cost of training much more 2-way classifiers.

Between the unique one-vs-others and all-vs-all methods, our tests indicate that the former appears to be more effective if only a single parameter dataset is available, and the latter is better for combining scores from multiple datasets. Overall, both methods appear to perform equally well.

Theoretically, the all-vs-all method has cleaner decision boundaries between all pairs of classes, but has larger noise due to the involvement of all possible pairs. Combining multiple votes on different parameter datasets reduces the noise, thus leading to more accurate predictions.

The unique one-vs-others method involves substantially fewer pairs of classes, thus less noise, at the false positive elimination step. This explains the high accuracy for a single parameter dataset; combining votes from more parameter dataset do reduce noise, but not as significantly as in the all-vs-all method. However, the decision boundaries used in the first step, the one-vs-others step, cannot be drawn as clean between one true class and the complementary "others" class. This is the fundamental limitation of this method.

The all-vs-all method was briefly mentioned in (Weston, 1998) and no improvement was found over the one-vs-others method.

8.2 Comparison between SVM and NN

Our results, as shown in Tables 6 and 7, demonstrate substantially higher accuracy achieved by SVM as compared to NN. As mentioned earlier, one of the pronounced features of NN is rather high false positive rates, due to higher noise levels in NN. This negatively impacts the prediction accuracy. The interesting point emerging from our study is that when scores of multiple parameter datasets are combined, accuracy for NN improves much more than for SVM, due to the significant reduction of noise in the case of NN. This indicates that the voting approach for NN is crucial to achieve high accuracy.

Another pronounced difference is computational efficiency. NN training typically converges slowly, whereas SVM training converges repidly, typically about 1-2 orders of magnitude faster than using NN. For this reason, some of the multi-way classification methods are only tested using SVM. The 10-fold cross-validation, dominated by the training of the 351*6*10=21060 two-way SVM classifiers shown in Table 7 took about 12 CPU hours on a Sun Ultra 5.

8.3 Effectiveness of Parameter Sets

The effectiveness of machine learning methods depends crucially on the feature vectors extracted from the protein sequence. Extensive testing of different classification methods on independent protein sets or by cross-validation showed that amino acid composition is the most effective parameter set, followed by the predicted secondary structure, and then hydrophobicity parameter sets. The numerical assessment is listed in Table 8. However, the best accuracy is obtained when scores of different parameter sets are combined together. This further confirms our earlier intuition in developing the feature extraction methods.

Parameter	SVM CV	SVM Ind-Test	NN Ind-Test	Avg
composition	32.7%	44.9%	20.5%	32.7%
secondary struc.	34.6	35.6	18.3	29.5
hydrophobicity	19.8	36.5	14.2	23.5
polarity	18.7	32.9	11.1	20.9
volume	17.2	35.0	13.4	21.8
polarizability	14.6	32.9	13.2	20.2

Table 8: Prediction accuracy Q for different parameter datasets. Both independent test (Ind-Test) and cross-validation (CV) are shown.

8.4 How many representatives does each fold need?

In Table 9, we show how the prediction accuracy of both cross-validation and independent test depends on the number of representative proteins in a fold. To gain sufficient statistics, we averaged those folds with representative proteins in ranges 7 - 9 (there are 12 folds in this range), 10-12, 13-16, and 29-30 (there are no folds with the number of proteins in the range 17-28). It is clear that as $N_{\rm rep}$ in each class increases, the accuracies increase steadily, to about 58-67% level for 29-30 representatives per class. This is quite consistent in cross-validations on training dataset; although there are a few exceptions in independent tests on classes with rather small number of proteins (7-9), where large fluctuations are expected.

	Cross Validation		Independent Test			
N_{rep}	AvA SVM	OvO NN	AvA SVM	uOvO SVM	OvO SVM	OvO NN
7- 9	31.1	13.4	51.4	46.6	37.6	34.1
10-12	38.9	18.3	42.1	42.4	41.5	30.1
13-16	50.3	27.8	57.2	54.8	46.5	41.7
29-30	67.0	58.1	74.5	62.4	53.9	59.9

Table 9: Effects on percentage accuracy Q due to number of representatives (N_{rep}) in each fold.

8.5 Effects of large number of folds

Prediction accuracy depends on the number of folds in the prediction system. To investigate this further, we studied 2-class and 8-class problems in addition to the 27-class problem and results are shown in Table 10. In 2-class problem, each fold is classified with each of other 26 folds in 2-way classification, and the prediction accuracy is averaged (2-way results). This is repeated for each fold. The 8-way classification involves folds 1, 20, 26, 32, 46, 51, 87, 110, which are chosen because each of the folds has 13 or more proteins.

For independent tests, the accuracy drops from 84.3% for 2-way classifications to 52.8% for the 8-way classification to 45.6% for the 27-way classification. The same trend is also apparent for cross-validations on either 8-way classification (63.7%) to 27-way classification (45.2%).

The reason for the steady drop in prediction accuracy is two-fold. First, as a general trend, the more classes are involved in a classification system, the more difficult it is to accurately assign a new query protein. Second, in our datasets, the number of representatives in each fold reduces very substantially, as explained in the previous section. From the significant drop in prediction accuracy shown in Table 9, we believe this factor is more important. Fortunately, this lack of representatives will be improved by the steady growth of the number of known proteins in databases.

Overall, for the 27-class dataset with relatively small number of representatives in each fold (many have 7 proteins), the prediction accuracy is around 50% (45% for CV, 56% for test). Although this accuracy level is not high, we note that for 27-class problem, a random prediction will have an accuracy of 1/27=3.7%.

Fold	2-way Test	8-way Test	27-way Test	8-way CV	27-way CV
1	91.7%	83.3%	83.3%	62.5%	71.1%
3	92.7	–	66.7	–	65.2
4	64.6	–	30.0	–	64.6
7	74.5	–	43.8	–	26.8
9	98.3	–	77.8	–	45.8
11	75.6	–	55.6	–	41.1
20	87.8	54.5	45.5	64.8	52.5
23	80.1	–	33.3	–	30.6
26	90.8	34.6	34.6	76.5	70.3
30	83.3	–	33.3	–	20.8
31	70.2	–	41.7	–	46.9
32	75.9	26.3	18.4	20.8	7.7
33	86.5	–	50.0	–	7.8
35	69.2	–	50.0	–	25.0
39	87.9	–	52.4	–	27.8
46	93.4	61.5	51.0	80.2	81.5
47	89.7	–	41.7	–	43.2
48	81.7	–	38.5	–	9.1
51	90.0	55.6	51.9	60.1	53.5
54	88.5	–	36.1	–	58.8
57	78.8	–	25.0	–	36.1
59	82.4	–	35.7	–	3.8
62	89.0	–	71.4	–	28.4
69	69.2	–	25.0	–	64.7
72	68.3	–	18.8	–	0.0
87	70.4	12.3	13.0	26.5	20.8
110	97.4	92.6	90.7	91.1	78.6
avg	84.3	52.8	45.6	63.7	45.2

Table 10: Dependency of SVM prediction accuracy Q_i, Q on the number of folds. Only composition parameter dataset is used.

8.6 Feedback to SCOP

Our study also shows that some folds are consistently recognized with high prediction accuracy: fold F9 (α: 4-helical cytokines), fold F26 (β: viral coat), fold F46 (α/β: TIM-barrel), fold F110 ($\alpha + \beta$: small inhibitors); while some other folds are consistently recognized with low accuracy: fold F23 (β: cuperedoxins), fold F59 (α/β: ribonuclear H-like motif), fold F72 ($\alpha + \beta$: β-grasp). These features are fairly persistent on different parameter datasets and combined datasets. They are also consistent with different discriminant methods (see Tables 4,5,7).

The biological characteristics of these folds are worth further examination, which could probably lead to better feature extraction methods for more accurate predictions, and could also provide feedback to improve the original SCOP classification database (e.g., split one difficult fold into several folds). Much remains to be explored here.

9 Conclusion

In this paper, we studied several important issues in protein fold recognition in the context of a large number of folds using discriminative methods, aided by the fast and highly accurate support vector machine. We studied the popular one-against-others method, and two new advanced methods: the unique one-vs-others method and the all-vs-all method. These advanced methods improved prediction accuracy substantially, at a higher but manageable computational cost.

Overall, recognition methods achieve 56% prediction accuracy on test proteins which have less than 35% sequence identity with proteins used in training (90% of those test proteins have less than 25% sequence identity with the training proteins, see Brenner, et. al. 1998, Fig. 6). Thus the fold recognition approach is a useful structure discovery method, complementary to BLAST type sequence-similarity based methods.

In present work, the recognition system simply predicts a fold for an input protein without associating it a numerical value to assess the reliability or confidence of the prediction. Since each protein is predicted with different reliability, such a reliability score is necessary for practical prediction systems. For example, a low reliability score for a new protein may signal that it does not belong to any folds in the system.

In this study, we also systematically investigated many important aspects of multi-class fold prediction, which will help to build a practical fold prediction system including about 600 folds in the SCOP database.

Acknowledgement. We thank Prof. M. Jordan, Drs. S. Holbrook, S. Mian, E. Xing, and H. Simon for SVM and fold recognition discussions. We also thank anonymous referees for useful comments and suggestions. We used SVM-Light software (http://ais.gmd.de/~thorsten/svm_light/). This work is supported by the Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences, and also by Office of Laboratory Policy and Infrastructure (through a LBNL LDRD grant), both of the U.S. Department of Energy under contract number DE-AC03- 76SF00098.

References

Baldi,P. and Brunak,S. (1998) *Bioinformatics : the machine learning approach.* Cambridge, Mass. MIT Press, c1998.

Baldi,P., Brunak,S., Chauvin,Y., Andersen,C. and Nielsen,H. (2000) Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16, 412-424.

Brenner, S.E., Chothia, C. and Hubbard, T.J. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. National Academy of Sciences*, 26, 6073-8.

Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C., Ares, Jr. M. and Haussler, D. (2000) Knowledge-based Analysis of Microarray Gene Expression Data. by using Support Vector Machines. *Proc. Natl Acad Sci.*, 97, 262-267.

Burges, C.J.C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge*

Discovery and Data Mining, 2, 1-43.

Chou, K.-C. and Zhang, C.T. (1995) Prediction of protein structural classes. *Critical Revi. Biochem. Mol. Biol.* 30, 275-349.

Craven, M.W., Mural, R.J., Hauser, L.J. and Uberbacher, E.C. (1995) Predicting protein folding classes without overly relying on homology. *ISMB*, 3, 98-106.

Dubchak, I., Muchnik, I., Holbrook, S.R. and Kim, S.H. (1995) Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl Acad Sci.* , 92, 8700-4.

Dubchak, I., Muchnik,I., Mayor,C., Dralyuk,I. and Kim,S.H. (1999) Recognition of a protein fold in the context of the Structural Classification of Proteins (SCOP) classification. *Proteins*, 35, 401-7.

Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998) *Biological Sequence Analysis.* Cambridge Press.

Hobohm, U., Scharf, M., Schneider, R. and Sander, C. (1992) Selection of a representative set of structures from the Brookhaven Protein Bank. *Protein Sci.*, 1, 409-417.

Hobohm, U. and Sander, C. (1994) Enlarged representative set of Proteins. *Protein Sci.*, 3, 522-524.

Holm, L. and Sander, C. (1999) Protein folds and families: sequence and structure alignments. *Nucleic Acids Research* , 27, 244-7.

Jaakkola, T., Diekhans, M. and Haussler, D. (1999) Using the Fisher kernel method to detect remote protein homologies. *ISMB*, 149-158

Joachims, T. (1998) Making large scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, ed. Scholkopf, B., Burges, C. and Smola, A. MIT Press.

Jones, D.T. (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Bio* , 287, 797-815.

Lo Conte, L., Ailey, B., Hubbard, T.J.P., Brenner, S. E., Murzin, A. G. and Chothia, C. (2000) SCOP: a Structural Classification of Proteins database *Nucleic Acids Res.*, 28, 257-259.

Osuna, E., Freund, R. and Girosi, F. (1997) An improved training algorithm for support vector machines. *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, 276-285.

Park, J., Karplus, K., Barret, C., Hughey, R., Haussler, D., Hubbard, T. and Chothia, C. (1998) Sequence Comparison Using Multiple Sequences Detect Three Times as Many Remote Homologues as Pairwise Methods. *J.Mol. Bio.*, 284, 1201-10.

Pearl, F.M., Lee, D., Bray, J.E., Sillitoe, I., Todd, A.E., Harrison, A.P., Thornton, J.M. and Orengo, C.A. (2000) Assigning genomic sequences to CATH. *Nucleic Acids Res.*, 28, 277-82.

Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Bio.*, 232, 584-599.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

Weston, J. and Watkins, C. (1998) Multi-class Support Vector Machines. Royal Holloway, Univ of London. Tech Report CSD-TR-98-04.